How to Use PowerCenter Web Services to Extend the Power of Data Integration

Abstract

This article shows how to extend PowerCenter’s ETL infrastructure to expose integrated data as web services. It discusses the benefits of enhancements and new features of web services in PowerCenter version 8.5 or later. It also provides sizing guidelines and sample performance results generated from web service testing within the Informatica labs.

Table of Contents

Executive Summary

For many years, IT organizations have used extract, transform, and load (ETL) technology for traditional batch-oriented data integration projects such as building data warehouses or migrating or consolidating data. PowerCenter’s rich metadata framework has provided the ability to reuse data transformation, integration, and cleansing rules seamlessly across the enterprise for different projects. High performance and scalability and rich connectivity to sources like relational databases, ERP systems, and hosted applications have proven PowerCenter to be ideal for managing large volumes of diverse data.

As enterprises have become more agile, the ability to exchange operational data between applications has become more critical. Data transformation, integration, and cleansing rules need to be applied to the data on the fly and provided to the consuming application in real-time.

Despite obvious shortcomings, some IT organizations have used enterprise application integration (EAI) techniques and J2EE application server technology to provide data to applications. Such technology lacks a strong metadata repository capability and integration with the ETL infrastructure. The integration logic in the EAI approach commonly involves hand coding, which is complex and very expensive to maintain. The transformation, integration, or cleansing rules developed for batch ETL are often similar to the ones needed for operational data integration. However, since the EAI technology is decoupled from the PowerCenter repository, the integration rules cannot be reused. Furthermore, the technology falls short when the volume of data grows. As a result, this approach to operational data integration is inflexible and not scalable.

Exposing integrated data as data services makes the ETL infrastructure more extensible and reusable across IT projects. Data services enable access, integration, and delivery of enterprise data throughout the enterprise and across corporate firewalls. Data services can be exposed in many modalities, including web services, SQL, and Java/C++ APIs.

Web services offer a method for data exchange among applications based on common SOA standards and protocols. The basic web service platform is XML plus HTTP. XML provides a language which can be used among different platforms and programming languages and the HTTP protocol is the most commonly used Internet protocol.

Extending PowerCenter's ETL infrastructure to expose integrated data as web services addresses the shortcomings of the EAI technology described above. Furthermore, PowerCenter can integrate data from external web service providers to allow it to participate fully in SOA architectures. The strength of ETL technology combines with web services standards to allow IT organizations to extend the reach of data integration from traditional data warehouse projects to operational applications on smaller budgets.

Business Use Case

Although the traditional enterprise data warehouse is seen as the information store for all enterprise data, it is often not capable of fulfilling the demands for more real-time information. The general practice within IT has been to implement the various data integration tasks for access, quality, manipulation and delivery using disparate tools tuned for different data latency modes and data volumes. This "hairball" of data integration logic has resulted in a lack of agility, low to no reusability, data inconsistency, poor data manageability, and complicated change management.

IT organizations are looking for a scalable flexible real-time data integration infrastructure that can form the basis of an enterprise information management framework for delivering business agility. This framework should help IT organizations to better manage the creation, management, manipulation, and delivery of enterprise data in a scalable, consistent, accurate, secure, and timely manner.

PowerCenter Solution for Use Case

Informatica's data services offering delivers a proven value to the end user for enabling large-volume data integration in the enterprise. At the heart of Informatica's data services platform is a high-performance engine for delivering scalable and sophisticated metadata-aware data integration services for access, cleansing, transformation, and delivery of data. The platform offers the flexibility of a variety of data delivery mechanisms, including web services.

Architecture

Web services provide a distributed computing platform that allows access to computational logic and data by other applications over the Internet and intranet using standard XML protocols and formats. Web services leverage open Internet standards:

- Web Services Description Language (WSDL) to describe the operations available in a web service.
- Universal Description, Discovery, and Integration (UDDI) to advertise and syndicate the web services.
- Simple Object Access Protocol (SOAP) to send and receive web service messages.
- Web Services Flow Language (WSFL) to define the web service processes.

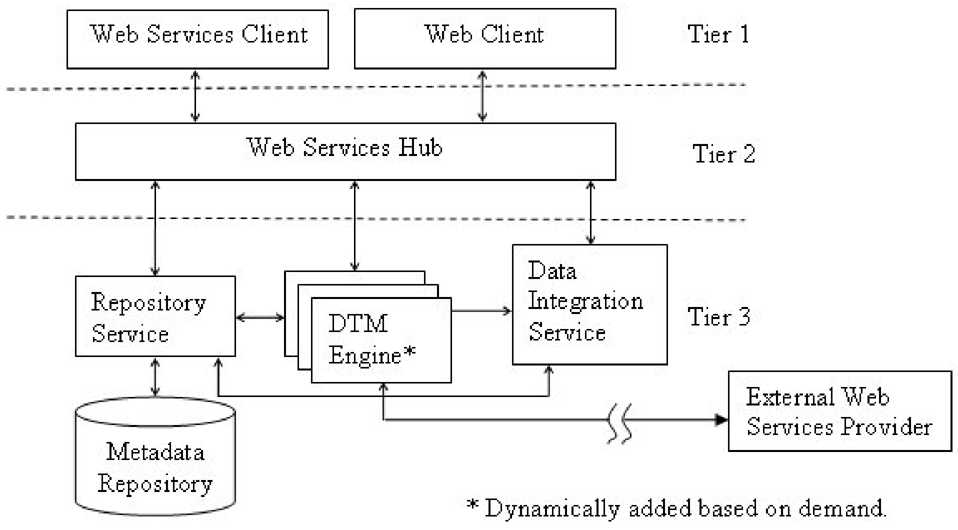
The use of standard XML protocol eliminates the interoperability issues of existing solutions, such as CORBA and DCOM, and makes web services platform, language, and vendor neutral. Many commercial software products have already provided or have started providing support for web service interface.

The PowerCenter web service solution is based on a three-tier architecture:

- Client
- Web Services Hub

- Data Integration

Figure 1. PowerCenter Web Services Architecture



Client

The first tier is the client tier and consists of two types of web clients:

- A web service client that accesses PowerCenter web services by sending web service requests and receiving web service responses in the form of SOAP messages over HTTP.
- A web client, typically a web browser, that accesses PowerCenter services and metadata by sending HTTP requests and receiving HTTP responses.

The web service client or web client can also send requests and receive responses through a secure HTTPS connection.

Web Services Hub

The second tier contains the Web Services Hub, the web service gateway to PowerCenter. The Web Services Hub uses the services in the third tier to serve the clients requests.

The following are some of the architectural features of the Web Services Hub:

- The Web Services Hub is built on a Tomcat web service container and runs as an independent process and hence resilient to failure of other Informatica services.
- The Web Service Hub processes requests and responses in blocks for improved throughput. It bundles multiple requests into data blocks and sends them to the DTM engine. Likewise, it receives multiple responses in data blocks from the DTM engine.
- Each connection from the Web Services Hub to the DTM engine is independent. Each data block of request or response is sent through an independent connection and is not queued behind other requests.
- The Web Services Hub limits the number of context switches by reusing active client threads to send the requests from different clients to the DTM engine.
- The Web Services Hub provides a range of web service operations that allow clients to access PowerCenter services and metadata. Operations such as startWorkflow and startTask allow clients to remotely start workflows and tasks. Operations such as getAllFolders and getWorkflowLog allow clients to access metadata from the repository and information on the workflow runs.

Data Integration

The third tier consists of the PowerCenter data integration components, including the Integration Service and the Repository Service. These application services handle any data transformation required in a request.

After the Web Services Hub receives and authenticates a web service request, the Web Services Hub sends the request to the Integration Service. The Integration Service starts a Data Transformation Manager (DTM) process to handle any data transformation required in a request. The Integration Service sends the results of the data transformation to the Web Services Hub which sends the response to the web service client.

You can set up a grid in the PowerCenter domain and configure workflows and sessions to run on the grid to improve performance and scalability. When you run a workflow on a grid, the Integration Service runs a DTM process on each available node of the grid. When you run a session on a grid, the Integration Service distributes session threads to multiple DTM processes on nodes in the grid.

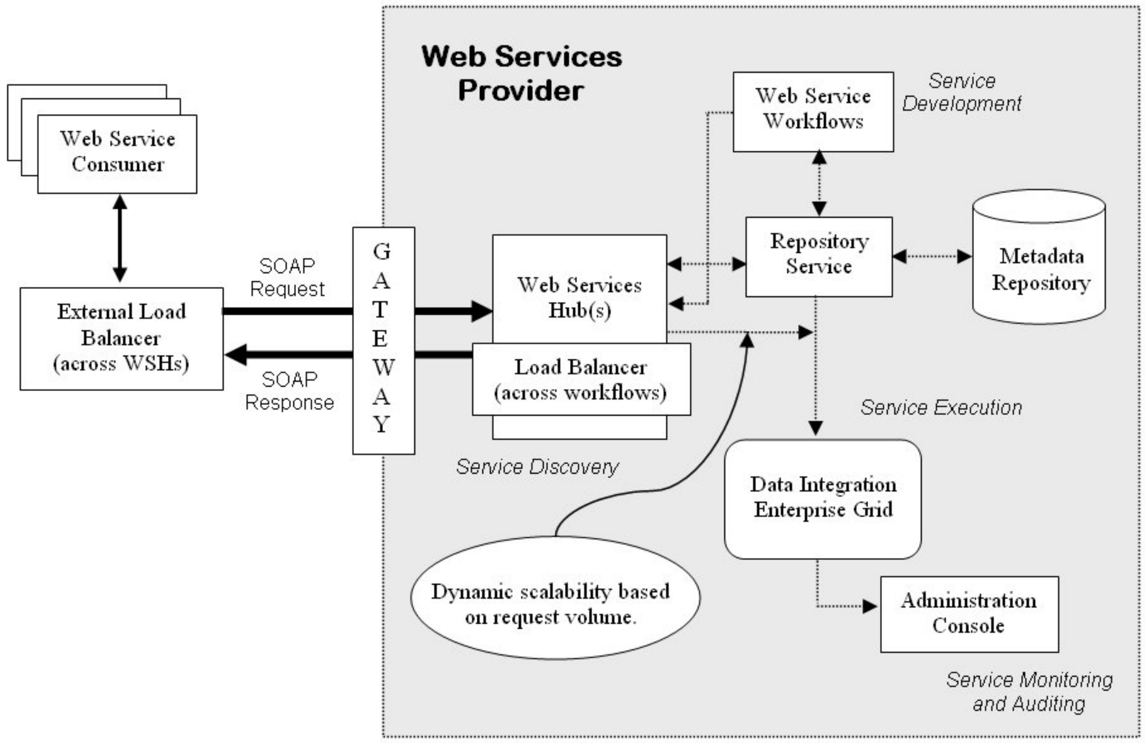
External Web Services

The PowerCenter web service solution also allows you to consume information from external web service providers for use in a PowerCenter data transformation. PowerCenter provides a Web Services Consumer transformation process as part of PowerExchange for Web Services that can act as a web service client within a PowerCenter workflow. When a workflow contains a Web Services Consumer transformation, the workflow can send a request to an external web service and use the response in the data transformation.

Deployment

Figure 2 shows a typical PowerCenter web services deployment:

Figure 2. PowerCenter Web Services Deployment



To ensure that the Web Services Hub does not become the bottleneck or a single point of failure, deploy an external load balancer to manage the requests going into the Web Services Hub. A load balancer can ensure that requests for a web service are balanced across multiple Web Services Hubs associated with the web service. The load balancer can also route requests through HTTP or HTTPS efficiently.

Security

The PowerCenter web services solution provides the following levels of security:

- **Transport layer security.** The SSL protocol provides security for the SOAP message transport. Using HTTPS ensures the integrity and confidentiality of SOAP messages and provides point-to-point security. To enable the transport layer security, set up a keystore file for the SSL keys and certificate. Then use the Administration Console to configure the Web Services Hub to use HTTPS.
- **Message layer security.** To provide security at the message level, configure a web service workflow as a protected web service. A protected web service requires a security token to be included in the request. The security token is a session ID that is generated when a web service client logs in with a user name and password. The session ID can be used for subsequent requests from the client. It expires after a period of client inactivity. The batch web service operations provided by the Web Service Hub require the client to log in and obtain a session ID.

The transport layer security ensures SOAP message integrity and confidentiality while the SOAP message layer security provides client authentication. To maximize security for PowerCenter web services, use protected web services and run them over a secure HTTPS connection.

Performance and Scalability

The PowerCenter web services solution is designed to be highly scalable and available to enable PowerCenter customers to manage high volumes of requests with minimum down time.

Power Center 8.5 includes a number of performance enhancements:

- The DTM engine can run with multiple partitions, allowing it to process multiple requests concurrently. To enable this feature, create multiple horizontal partitions when you design a mapping.
- A Web Services Hub can dynamically launch multiple DTM processes to process requests concurrently, based on the request load and expected service time. The Web Service Hub monitors the quality of service for each web service workflow. You can set the maximum number of DTM processes that can be launched and the service time threshold at which a new DTM process will launched. When the service time threshold is crossed, the Web Services Hub launches a new DTM process to handle requests.
- To identify performance bottlenecks, the Web Services Hub monitors the minimum, maximum, and average processing time per request (including and excluding Web Services Hub processing), the number of lost connections, and the percentage of partitions used. The Web Services Hub uses the statistics to determine when a new DTM process needs to be launched to keep data transformation performance at an optimum level.
- If the Web Services Hub is the performance bottleneck, you can create multiple Web Services Hubs and improve performance in the following ways:
    - **Associate multiple Web Services Hubs with the same set of web services workflows.** When multiple Web Services Hubs can run the same web services, clients can access any of the Web Services Hubs to run a web service.
    - **Associate each Web Services Hub with a different set of web services workflows.** When each Web Services Hub runs a different set of web services, clients must access a specific Web Services Hub to run a web service. You can balance the load for each Web Services Hub by associating the Web Services Hub with web services that vary in levels or times of demand.

  To further improve performance, use a software or hardware load balancer to manage the volume of requests sent to the Web Services Hubs.

- Multiple Web Services Hubs can be associated with a Repository Service in a domain. You can set up multiple Web Services Hubs to run a web service workflow. This means that if one Web Service Hub fails, another Web Service Hub can run the web service workflow. When you set up a load balancer to manage request load across multiple Web Services Hubs, the URL of the load balancer is the service access point for all managed Web Services Hubs. The web service clients access the load manager URL to run any of the web service workflows associated with the managed Web Services Hubs.

Performance Tuning Parameters

You can configure the following properties to enable the Web Services Hub to perform at an optimum level for the required workload:

- MaxISConnections. Maximum number of connections that can be open at one time from the Web Services Hub to the Integration Service.
- MaxConcurrentRequests. Maximum number of request processing threads allowed, which determines the maximum number of simultaneous requests that can be handled. Before you set this parameter, check the memory available on the machine that hosts the Web Services Hub. This parameter allocates 64KB of memory to each request.
- MaxQueueLength. Maximum number of requests that can be kept in queue when the Web Services Hub reaches the maximum concurrent request limit and all possible request processing threads are in use. Any request received when the queue is full is rejected.

Use the PowerCenter Administration Console to configure these advanced Web Services Hub properties.

Benefits of Upgrading from PowerCenter 8.1.1

When you upgrade to PowerCenter version 8.5 or later, you can take advantage of the following enhancements and new features:

- Improved performance and reliability. PowerCenter 8.5 has made tremendous improvements in web service performance. The following charts show the comparison of service time and throughput performance for PowerCenter 8.1.1 and 8.5.

Figure 3. Average Service Time for PowerCenter 8.1.1 and 8.5

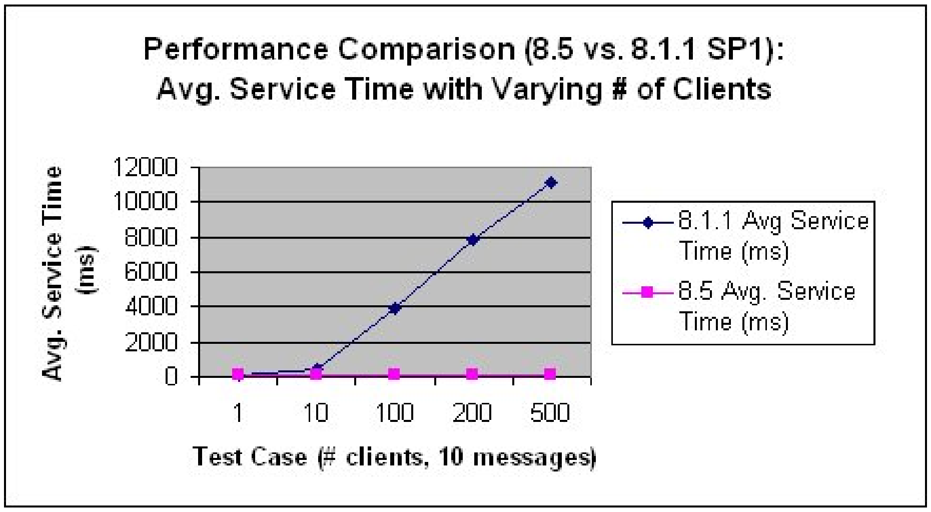
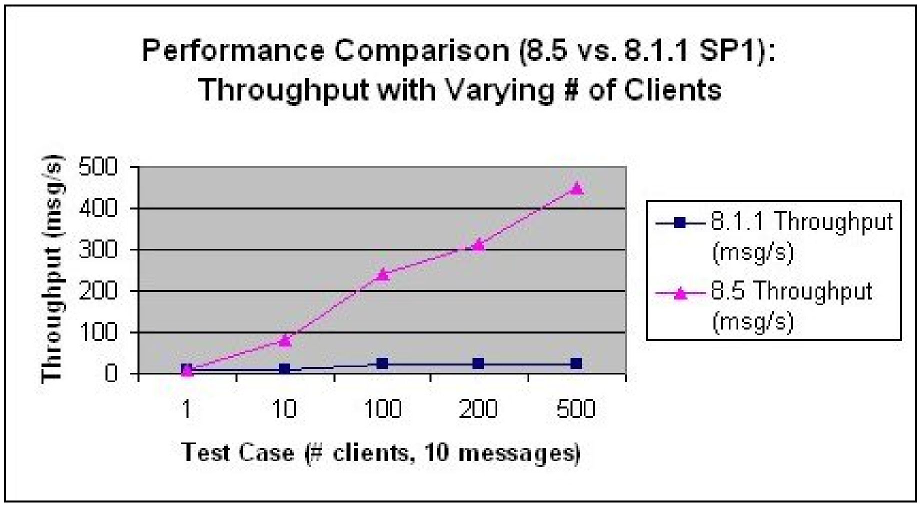


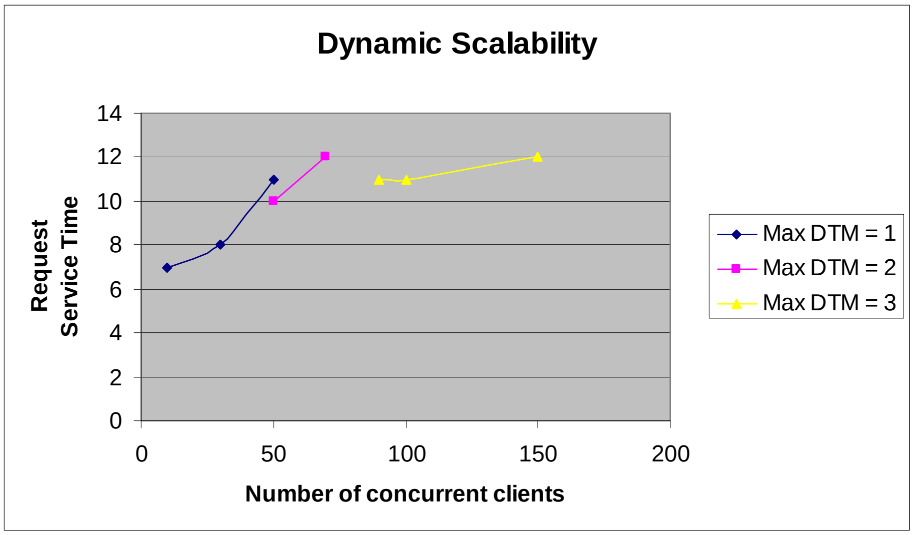
Figure 4. Average Throughput for PowerCenter 8.1.1 and 8.5



- **Dynamic scalability.** The Web Services Hub monitors web services performance and dynamically starts new DTM processes to handle an increase in web services requests. A larger number of DTM processes increases the number of client requests that can be processed within the target service time. When the load decreases, the Web Services Hub shuts down the additional DTM. This means that, at any given time, the DTM engine's usage of system resources is optimized and reflects the overall load on the system.

  Figure 5 shows that a larger number of DTM processes running concurrently can handle more client requests in less time:

Figure 5. Average Service Time for Single and Multiple DTM Processes



- **Multiple Web Service Hubs associated with a repository.** When you associate multiple Web Services Hubs with one repository, multiple Web Services Hubs can run the same web services concurrently. Distributing the service request load across multiple Web Services Hubs optimizes the performance of web services.
- **New methods to create web service mappings.** You can create a web service mapping by defining the source and target manually or basing the source and target definitions on existing relational or flat file sources and targets. You can also create a service mapping based on a reusable transformation or a mapplet.

- **Web service reports.** On the Administration Console, you can run a report on the activities of a Web Services Hub and the web services running on the Web Services Hub. You can view statistics on the requests received by the Web Services Hub and the average time it took to process messages.
- **Try-It application.** You can use the Try-It application to test an operation in a web service published on the Web Services Hub console. Provide the request as a SOAP message or as parameter values and then run the web service.

Sizing Guidelines

The service time threshold parameter determines the expected length of time for an average service request to be processed. It determines the maximum response time acceptable for an average service request. In most situations, the service time threshold must be measured in sub-seconds.

If the service load increases and the response time exceeds the service time threshold, the Web Services Hub launches additional workflow instances to handle the load and to maintain the service time threshold.

Each additional workflow instance consumes additional memory and resources. The complexity of the web service mapping determines how much additional memory and resources a workflow instance consumes.

Sizing for PowerCenter web services depends on the following factors:

- Complexity of the Web Service Mapping
- Service Time threshold
- Number of concurrent clients

Sample Performance Results

The following example shows performance results generated from web service testing within the Informatica labs. The test environment consisted of two nodes: one node running the Web Services Hub and another node running the Integration Service. The two nodes were running on an isolated 1Gbps network.

Each node had the following specification:

- 2 CPU (clock speed 1.79GHz; AMD Opteron™ Processor 844; 64-bit)
- 4GB RAM
- Linux 2.6.9-34.0.1.ELsmp

Note: You can also use one machine to host the Web Service Hub and the PowerCenter services. It is not necessary to host the services on separate machines.

Mapping

The mapping used for the test consisted of a simple pass-through mapping with a 2KB payload. This mapping was chosen so that the capacity of the Integration Service node does not affect throughput. As the mapping logic gets more complex, the Integration Service node may require additional hardware.

Response Time

The following table shows the average response time generated during the testing. Note that, with a larger number of clients, the response time benefits from having messages bundled together.

Number of Clients	Average Response Time (ms)
1	111
10	105

Number of Clients	Average Response Time (ms)
100	86
200	77
500	72

Scalability

The following table shows the throughput generated during the testing:

Number of Clients	Throughput (msgs/sec)
1	8.6
10	82.9
100	239.8
200	313.7
500	448.5

Authors

Kiran Mehta
Director, Research and Development

Raymond To
Development Manager

Marissa Johnston
Principal Technical Writer